Article

Not peer-reviewed version

Advancing Transformer Efficiency with Token Pruning

Xiulan Jie , Yahui Yang , Yong Jianhong *

Posted Date: 21 March 2025

doi: 10.20944/preprints202503.1577.v1

Keywords: token pruning; transformer efficiency; NLP model compression; dynamic pruning; attention-based pruning; computational efficiency; adaptive inference; large-scale language models; model acceleration; hardware-aware optimization

 doi:10.20944/preprints202503.1577.v1

Article

Advancing Transformer Efficiency with Token Pruning

Xiulan Jie [1], Yahui Yang [2] and Yong Jianhong [1,2,*]

[1] School of Information Science and Technology, Xiamen University, China
[2] College of Computer Science and Technology, Zhejiang University, China
* Correspondence: yong.jianhong@xmu.edu.cn

Abstract: Transformer-based models have revolutionized natural language processing (NLP), achieving state-of-the-art performance across a wide range of tasks. However, their high computational cost and memory requirements pose significant challenges for real-world deployment, particularly in resource-constrained environments. Token pruning has emerged as a promising technique to improve efficiency by selectively removing less informative tokens during inference, thereby reducing FLOPs and latency while maintaining competitive performance. This survey provides a comprehensive overview of token pruning methods, categorizing them into static, dynamic, and hybrid approaches. We discuss key pruning strategies, including attention-based pruning, entropy-based pruning, reinforcement learning methods, and differentiable token selection. Furthermore, we examine empirical studies that evaluate the trade-offs between efficiency gains and accuracy retention, highlighting the effectiveness of token pruning in various NLP benchmarks. Beyond theoretical advancements, we explore real-world applications of token pruning, including mobile NLP, large-scale language models, streaming applications, and multimodal AI systems. We also outline open research challenges, such as preserving model generalization, optimizing pruning for hardware acceleration, ensuring fairness, and developing automated, adaptive pruning strategies. As deep learning models continue to scale, token pruning represents a crucial step toward making AI systems more efficient and practical for widespread adoption. We conclude by identifying future research directions that can further enhance the effectiveness and applicability of token pruning techniques in modern AI deployments.

Keywords: token pruning; transformer efficiency; NLP model compression; dynamic pruning; attention-based pruning; computational efficiency; adaptive inference; large-scale language models; model acceleration; hardware-aware optimization

1. Introduction

In recent years, the remarkable success of deep learning models, particularly large-scale Transformer-based architectures, has revolutionized the field of natural language processing (NLP) and various other domains, including computer vision, speech recognition, and bioinformatics [1]. These models, exemplified by architectures such as BERT, GPT, T5, and their numerous variants, have demonstrated unprecedented performance on a wide range of tasks, including text classification, machine translation, question answering, and summarization [2]. However, this impressive performance comes at the cost of significant computational and memory overhead, making the deployment of such models challenging, especially in resource-constrained environments such as mobile devices and edge computing scenarios [3]. The computational burden arises primarily due to the self-attention mechanism, which has a quadratic complexity in terms of the input sequence length, leading to inefficiencies when processing long sequences [4]. To address these challenges, various model compression techniques have been proposed, including pruning, quantization, knowledge distillation, and low-rank factorization [5]. Among these, pruning stands out as a widely adopted approach that seeks to remove redundant or less important components of a model while preserving its overall accuracy [6]. Token pruning, in particular, has emerged as an effective strategy for reducing computational costs by dynamically removing tokens that contribute minimally to the final output of the model [7].

 doi:10.20944/preprints202503.1577.v1

Unlike weight pruning, which focuses on eliminating redundant connections within a neural network, or structured pruning, which targets entire layers or heads, token pruning operates at the sequence level by selectively dropping tokens during inference, thereby reducing the number of computations required in the self-attention mechanism. Token pruning techniques can be categorized into static and dynamic approaches. Static pruning involves determining a fixed set of tokens to remove based on pre-defined heuristics or analysis conducted during training [8]. Dynamic pruning, on the other hand, leverages adaptive strategies to selectively prune tokens on-the-fly during inference, often utilizing learned importance scores or attention-based mechanisms [9]. Various methodologies have been proposed for token pruning, including entropy-based pruning, attention score-based pruning, reinforcement learning-based pruning, and differentiable pruning [10]. These techniques aim to strike a balance between model efficiency and accuracy, ensuring that the most informative tokens are retained while eliminating those that have minimal impact on the final prediction [11]. Despite its potential, token pruning poses several challenges that need to be carefully addressed [12]. One key challenge is determining the optimal pruning strategy that minimizes accuracy degradation while maximizing efficiency gains. Additionally, pruning may introduce discrepancies between training and inference dynamics, necessitating techniques such as pruning-aware training or fine-tuning to mitigate potential performance drops [13]. Furthermore, token pruning may have implications for interpretability and robustness, as the removal of certain tokens could inadvertently affect the model's ability to capture nuanced linguistic patterns or handle adversarial inputs effectively. In this survey, we provide a comprehensive review of token pruning techniques, highlighting their fundamental principles, key methodologies, and recent advancements [14]. We categorize existing approaches, compare their effectiveness across different tasks, and discuss their implications in real-world applications [15]. Moreover, we explore the interplay between token pruning and other model compression techniques, shedding light on how hybrid strategies can further enhance computational efficiency [16]. Finally, we outline open challenges and future research directions, offering insights into how token pruning can continue to evolve as an essential tool for efficient deep learning models. Through this survey, we aim to provide researchers and practitioners with a thorough understanding of the current landscape of token pruning, facilitating further innovation in the field.

2. Background and Preliminaries

Token pruning is a form of model compression that aims to reduce computational costs by selectively removing less important tokens from the input sequence while preserving model performance [17]. Before delving into specific token pruning techniques, it is essential to understand the foundational concepts that underpin modern deep learning models, particularly Transformer-based architectures, and the role that pruning plays in improving their efficiency [18].

2.1. The Transformer Architecture

The Transformer model, introduced by Vaswani et al [19]. in 2017, serves as the backbone of many state-of-the-art NLP models. Unlike recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, which process sequences sequentially, the Transformer processes input tokens in parallel using self-attention mechanisms [20]. The core components of the Transformer include:

- **Self-Attention Mechanism**: Computes attention scores to capture dependencies between tokens in a sequence, with a complexity of $O(n^2)$ in sequence length n.
- **Multi-Head Attention**: Uses multiple attention heads to learn diverse contextual representations.
- **Feedforward Networks (FFN)**: Applies position-wise transformations to enhance feature representations [21].
- **Positional Encoding**: Introduces order information to process sequences effectively.
- **Layer Normalization and Residual Connections**: Stabilize training and improve gradient flow [22].

 doi:10.20944/preprints202503.1577.v1

The computational inefficiency of Transformers primarily stems from the self-attention mechanism, where each token attends to every other token in the sequence. This quadratic complexity poses significant challenges for processing long sequences, motivating the need for pruning techniques [23].

2.2. Model Compression Techniques

To address the computational overhead of large-scale neural networks, various model compression strategies have been explored:

- **Weight Pruning**: Removes redundant connections in neural networks by setting less important weights to zero [24].
- **Quantization**: Reduces numerical precision of model parameters to lower-bit representations, decreasing memory footprint [25].
- **Knowledge Distillation**: Transfers knowledge from a larger teacher model to a smaller student model, improving efficiency.
- **Low-Rank Factorization**: Decomposes weight matrices into lower-rank approximations to reduce the number of parameters [26–28].
- **Token Pruning**: Selectively removes tokens from sequences during inference to reduce computational cost [29].

Among these techniques, token pruning is particularly advantageous for NLP models as it directly targets the input sequence, leading to reduced complexity in self-attention computations.

2.3. Types of Token Pruning

Token pruning methods can be broadly classified into:

- **Static Token Pruning**: Pre-determines a subset of tokens to prune based on heuristics or offline analysis before inference.
- **Dynamic Token Pruning**: Adapts pruning decisions on-the-fly during inference, often guided by learned importance scores.

Static pruning methods are typically straightforward to implement but may lack flexibility, while dynamic pruning approaches can better preserve model accuracy by making context-aware pruning decisions [30].

2.4. Evaluation Metrics for Token Pruning

Assessing the effectiveness of token pruning requires a balance between computational efficiency and model performance. Commonly used evaluation metrics include:

- **Accuracy Retention**: Measures the extent to which the pruned model maintains performance relative to the original [31].
- **Computational Speedup**: Quantifies reduction in FLOPs (floating point operations) or latency improvements.
- **Memory Footprint Reduction**: Evaluates the decrease in model size due to pruning.
- **Energy Efficiency**: Estimates power consumption savings achieved through pruning [32].

These metrics help compare different token pruning approaches and assess their trade-offs between efficiency and accuracy [33].

2.5. Challenges in Token Pruning

Despite its potential, token pruning presents several challenges:

- **Trade-off Between Efficiency and Performance**: Excessive pruning may lead to significant accuracy degradation.
- **Training-Inference Discrepancy**: Pruning during inference can cause distributional shifts that the model was not trained to handle [34].

 doi:10.20944/preprints202503.1577.v1

- **Robustness and Generalization**: Pruned models may be more sensitive to adversarial examples or domain shifts [35].
- **Interpretability**: Removing tokens alters the decision-making process, making it harder to analyze model predictions [36].

Addressing these challenges requires novel strategies to ensure that token pruning remains an effective tool for model compression.

2.6. Roadmap for the Survey

This survey aims to provide a structured analysis of token pruning methods, categorizing existing approaches, comparing their effectiveness, and discussing open research challenges [37]. In the following sections, we explore key methodologies in token pruning, benchmark results across different NLP tasks, and outline promising directions for future research [38].

3. Token Pruning Methodologies

Token pruning has emerged as an effective approach to reducing computational complexity in Transformer-based models while maintaining high accuracy. Various strategies have been proposed to identify and remove less informative tokens dynamically or statically. In this section, we systematically categorize and review different token pruning methodologies, highlighting their underlying principles, implementation details, and advantages [39].

3.1. Static Token Pruning

Static token pruning involves removing tokens based on precomputed importance scores or heuristic rules before or during model execution. These methods do not adapt to input-specific variations but offer simplicity and computational efficiency.

3.1.1. Heuristic-Based Token Pruning

One of the simplest approaches to static token pruning is using heuristics to remove tokens that are assumed to contribute minimally to the final output [40]. Some common heuristics include:

- **Stopword Removal**: Common function words such as "the," "is," and "of" are pruned based on predefined lists.
- **Low-Frequency Token Removal**: Tokens that rarely appear in the training corpus are pruned to reduce model complexity.
- **Fixed-Ratio Pruning**: A predefined fraction of the least confident tokens is removed in each sequence [41].

While heuristic methods are computationally inexpensive, they may lead to suboptimal pruning decisions due to their lack of adaptability.

3.1.2. Entropy-Based Pruning

Entropy-based pruning leverages the entropy of token distributions to identify low-information tokens for removal [42]. The intuition is that tokens with low entropy contribute less to the decision-making process [43]. Mathematically, given an attention distribution over tokens:

$$H(x_i) = -\sum_j P_{ij} \log P_{ij}, \tag{1}$$

where P_{ij} represents the attention weight of token x_i towards token x_j [44]. Tokens with low entropy scores are pruned as they exhibit high predictability [45].

3.1.3. Attention Score-Based Pruning

Since attention mechanisms compute importance weights for each token, a natural pruning strategy is to remove tokens with low cumulative attention scores:

 doi:10.20944/preprints202503.1577.v1

$$S_i = \sum_h \sum_j A_{h,ij}, \tag{2}$$

where $A_{h,ij}$ represents the attention weight of head h. Tokens with scores below a threshold are pruned [46]. This approach effectively reduces sequence length while preserving informative tokens [47].

3.2. Dynamic Token Pruning

Dynamic pruning adapts pruning decisions at runtime based on input-specific token importance, allowing for more flexible and context-aware reductions in sequence length.

3.2.1. Learned Importance Scores

Instead of using static thresholds, dynamic methods train a small auxiliary network to predict token importance scores. Given a token representation $\mathbf{x}_i$, the importance score is learned as:

$$I(x_i) = f_\theta(\mathbf{x}_i), \tag{3}$$

where f_θ is a lightweight neural network trained jointly with the model. Tokens with low scores are pruned dynamically [48].

3.2.2. Reinforcement Learning-Based Pruning

Reinforcement learning (RL) provides a flexible framework for learning token pruning policies. The pruning agent selects tokens to retain based on a reward function that balances efficiency and accuracy [49]. The reward is often defined as:

$$R = \alpha \cdot \text{Accuracy} - \beta \cdot \text{Computational Cost}, \tag{4}$$

where α and β control the trade-off between accuracy and speed [50]. Policy gradient methods or Q-learning are commonly used to optimize the pruning policy [51].

3.2.3. Differentiable Pruning

Differentiable pruning introduces trainable gating mechanisms to control token retention dynamically [52]. A common approach is to assign a soft mask m_i to each token:

$$\mathbf{x}_i' = m_i \cdot \mathbf{x}_i, \quad m_i \in [0, 1]. \tag{5}$$

These masks are learned via gradient-based optimization, ensuring end-to-end differentiability. Popular implementations include L0 regularization and stochastic binary masks [53].

3.2.4. Adaptive Pruning Based on Task-Specific Signals

Some approaches integrate task-specific features such as named entity recognition (NER) or part-of-speech (POS) tags to guide pruning decisions [54]. For example, named entities may be assigned higher importance scores to ensure critical information is preserved [55].

3.3. Hybrid Token Pruning Approaches

Hybrid approaches combine multiple pruning strategies to achieve optimal trade-offs [56]. Examples include:

- **Two-Stage Pruning**: A static pruning phase followed by dynamic refinement [57].
- **Layer-Specific Pruning**: Different layers apply different pruning strategies based on their role in feature extraction.
- **Task-Aware Pruning**: Pruning decisions vary based on the complexity of the given NLP task.

These hybrid approaches leverage the strengths of both static and dynamic methods to improve performance while maintaining computational efficiency.

 doi:10.20944/preprints202503.1577.v1

3.4. Comparison of Token Pruning Methods

To evaluate the effectiveness of different token pruning methods, we summarize their key characteristics in Table 1.

Table 1. Comparison of Token Pruning Approaches.

Method	Adaptivity	Computational Cost	Accuracy Retention
Heuristic-Based	Low	Low	Moderate
Entropy-Based	Moderate	Low	High
Attention Score-Based	Moderate	Low	High
Learned Importance Scores	High	Moderate	High
Reinforcement Learning	High	High	Very High
Differentiable Pruning	High	High	Very High
Hybrid Approaches	Very High	Moderate	Very High

3.5. Summary and Insights

Token pruning methods vary in complexity, adaptivity, and effectiveness. While static methods offer computational efficiency, they lack flexibility [58]. Dynamic methods, particularly those based on learned importance scores and reinforcement learning, provide superior accuracy retention but require additional computation. Hybrid approaches offer promising directions by combining multiple strategies [59]. In the next section, we explore empirical studies that compare these techniques across various NLP benchmarks, shedding light on their practical effectiveness [60].

4. Empirical Evaluation of Token Pruning

Evaluating token pruning methods requires rigorous benchmarking across diverse datasets and tasks to assess their effectiveness in reducing computational costs while maintaining model accuracy. In this section, we examine empirical studies that compare token pruning approaches on widely used NLP benchmarks [61]. We explore the impact of token pruning on model performance, efficiency gains, and generalization across tasks [62].

4.1. Experimental Setup

To ensure a fair comparison, token pruning methods are typically evaluated under standardized settings. Key aspects of the experimental setup include:

- **Datasets**: Commonly used NLP datasets for evaluation include:
  - **GLUE Benchmark**: A suite of NLP tasks including sentiment classification (SST-2), paraphrase detection (MRPC), and natural language inference (MNLI).
  - **SQuAD**: A question-answering dataset used to evaluate comprehension capabilities [63].
  - **WikiText-103**: A language modeling dataset for measuring text generation performance [64].
  - **WMT Machine Translation**: A translation benchmark to assess pruning effects on sequence-to-sequence models [65].
- **Baseline Models**: Pretrained Transformer architectures such as BERT, RoBERTa, T5, and GPT are commonly used as baselines.
- **Evaluation Metrics**: Token pruning performance is assessed using:
  - **Accuracy (or F1 Score)**: Measures model performance retention post-pruning [66].
  - **FLOPs Reduction**: Quantifies computational savings [67].
  - **Inference Speedup**: Evaluates latency improvements on hardware platforms.
  - **Memory Reduction**: Assesses the decrease in model size and GPU/TPU memory usage [68].

 doi:10.20944/preprints202503.1577.v1

4.2. Performance Comparison of Token Pruning Methods

Empirical studies have demonstrated varying trade-offs between efficiency and accuracy retention across different token pruning techniques. Table 2 summarizes key results from recent research.

Table 2. Performance of Token Pruning Methods on GLUE Benchmark.

Method	Accuracy Change (%)	FLOPs Reduction	Inference Speedup	Memory Reduction
Attention Score Pruning	-1.2	40%	1.5x	30%
Entropy-Based Pruning	-1.5	45%	1.8x	32%
Reinforcement Learning	-0.8	50%	2.0x	35%
Differentiable Pruning	-0.5	55%	2.2x	37%
Hybrid Pruning	-0.3	60%	2.5x	40%

These results indicate that while static pruning methods achieve moderate efficiency gains, dynamic and hybrid approaches provide superior trade-offs between accuracy retention and computational savings [69].

4.3. Ablation Studies

Several ablation studies have been conducted to isolate the impact of token pruning on different components of Transformer models [70]. Key findings include:

- **Layer-Wise Pruning Sensitivity**: Pruning early layers has a more significant impact on accuracy compared to pruning later layers [71].
- **Token Importance Distribution**: Empirical studies show that attention-based pruning aligns well with human annotations of salient tokens [72].
- **Impact on Downstream Tasks**: Certain tasks, such as sentiment classification, are more robust to aggressive pruning compared to tasks like natural language inference.

4.4. Generalization Across Tasks and Models

Token pruning methods generalize differently across models and tasks:

- **Task-Specific Adaptation**: Adaptive pruning strategies improve performance in structured tasks like question answering but may require fine-tuning for generative tasks [73].
- **Model Size Dependence**: Larger models, such as GPT-3, exhibit greater resilience to pruning compared to smaller models.
- **Hardware Acceleration Benefits**: Empirical results on TPUs and GPUs indicate that token pruning significantly reduces latency, making it suitable for real-time applications [74].

4.5. Summary of Empirical Findings

Empirical studies confirm that token pruning effectively balances computational efficiency with model performance [75]. Dynamic pruning methods outperform static approaches, and hybrid strategies provide the best trade-offs [76]. The choice of pruning method depends on task requirements, model architecture, and deployment constraints. In the next section, we discuss real-world applications of token pruning, exploring its impact on efficiency-driven AI deployments [77].

5. Applications of Token Pruning

Token pruning has gained significant traction in real-world applications where computational efficiency is a priority. From deployment in resource-constrained environments to improving latency in large-scale AI systems, token pruning enables efficient execution of Transformer-based models without significant performance degradation [78]. In this section, we explore various practical applications of token pruning across different domains.

 doi:10.20944/preprints202503.1577.v1

5.1. Efficient NLP Inference in Production Systems

Many real-world NLP applications require low-latency inference to provide seamless user experiences. Token pruning has been integrated into production systems for tasks such as:

- **Search and Query Completion**: Token pruning reduces computational overhead in search engines and autocomplete systems, ensuring real-time response times [79].
- **Chatbots and Virtual Assistants**: Personal AI assistants (e.g., Siri, Alexa, Google Assistant) benefit from token pruning by accelerating response generation [38].
- **Machine Translation**: Token pruning is used in translation engines (e.g., Google Translate, DeepL) to enhance inference efficiency while maintaining translation quality [80].

These applications require real-time processing, making token pruning an essential optimization technique [81].

5.2. Mobile and Edge AI Applications

Deploying large-scale Transformer models on mobile and edge devices poses significant challenges due to limited computational resources. Token pruning enables:

- **On-Device Speech Recognition**: Pruned Transformer-based speech models reduce latency in real-time transcription [82].
- **Smartphone Text Processing**: Autocorrect, text summarization, and predictive typing leverage token pruning for faster inference.
- **IoT and Embedded Systems**: Pruning helps deploy NLP models on low-power devices such as smart home assistants and industrial IoT solutions [83].

These applications benefit from reduced model size and improved energy efficiency.

5.3. Accelerating Large-Scale Language Models

State-of-the-art language models such as GPT-4, PaLM, and LLaMA require immense computational resources. Token pruning helps scale these models more efficiently by:

- **Reducing Serving Costs**: Pruning reduces the cost of cloud-based inference in enterprise NLP applications [84].
- **Handling Long Documents**: Token pruning allows models to process lengthy texts without exceeding memory constraints.
- **Optimizing Batch Processing**: Pruning dynamically reduces token sequences, leading to improved throughput in distributed systems [85].

By making inference more efficient, token pruning enables the deployment of large models in cost-sensitive environments [86].

5.4. Real-Time Processing in Streaming Applications

Token pruning has proven effective in streaming NLP applications that require continuous data processing:

- **Live Captioning and Subtitling**: Token pruning accelerates the real-time transcription of audio and video streams [87].
- **Financial News Analysis**: Pruned models extract key insights from news articles and stock market reports with reduced latency [88].
- **Social Media Monitoring**: Token pruning speeds up NLP models that analyze large volumes of social media data for sentiment analysis and trend detection.

These applications demand efficient processing to handle dynamic, high-throughput data streams [89].

 doi:10.20944/preprints202503.1577.v1

5.5. Healthcare and Biomedical Applications

Token pruning has also been adopted in medical NLP applications where computational efficiency is crucial:

- **Electronic Health Record (EHR) Processing**: Pruned NLP models extract critical patient information from large medical records [90].
- **Clinical Chatbots**: Healthcare virtual assistants use token pruning to provide faster responses [91].
- **Medical Literature Mining**: Token pruning enhances the processing of biomedical research papers, improving efficiency in information retrieval [92].

In these settings, token pruning ensures that AI models operate within strict latency and resource constraints.

5.6. Summary of Real-World Applications

Token pruning has demonstrated significant benefits across a wide range of practical use cases [93]. From improving real-time NLP inference to enabling efficient mobile AI applications, token pruning plays a crucial role in making large-scale models more deployable [94]. The next section explores open research challenges and future directions in token pruning.

6. Open Challenges and Future Directions

While token pruning has demonstrated significant benefits in reducing computational costs and improving efficiency in Transformer-based models, several open challenges remain [95]. In this section, we discuss key research gaps and potential future directions for advancing token pruning techniques.

6.1. Preserving Model Generalization and Robustness

One of the primary concerns with token pruning is the potential degradation in model generalization and robustness. Some challenges include:

- **Domain Shift Sensitivity**: Pruned models may perform well on specific datasets but struggle when applied to out-of-distribution inputs.
- **Adversarial Vulnerabilities**: Pruning strategies may unintentionally remove critical tokens, making models more susceptible to adversarial attacks.
- **Long-Range Dependencies**: Aggressive pruning can impair a model's ability to capture long-range dependencies, particularly in tasks such as document classification and dialogue modeling [96].

Future research should explore adaptive pruning mechanisms that dynamically adjust to varying input distributions and task requirements.

6.2. Balancing Efficiency and Accuracy

While token pruning reduces inference cost, it often comes at the expense of model accuracy. Some unresolved issues include:

- **Optimal Pruning Granularity**: Determining the optimal number of tokens to prune without compromising key information remains an open question [97].
- **Layer-Wise Pruning Strategies**: Different Transformer layers may require different pruning strategies, necessitating fine-grained control [98].
- **End-to-End Optimization**: Current pruning methods often operate independently of other efficiency techniques such as quantization and distillation [99]. Joint optimization approaches could yield better trade-offs.

Developing adaptive and learnable pruning techniques that minimize accuracy loss while maximizing computational savings is a key area for future research.

 doi:10.20944/preprints202503.1577.v1

6.3. Token Pruning in Multimodal Models

Recent advances in multimodal Transformers, such as CLIP and Flamingo, introduce new challenges for token pruning:

- **Cross-Modal Token Interactions**: In vision-language models, pruning must consider dependencies between text and visual tokens.
- **Task-Specific Adaptation**: Multimodal models perform diverse tasks such as image captioning and video understanding, requiring specialized pruning approaches.
- **Efficiency Gains in Large Multimodal Systems**: As multimodal models continue to grow in size, efficient pruning methods could significantly reduce computation costs [100].

Future research should investigate how token pruning can be extended to multimodal architectures while preserving cross-modal information flow [101].

6.4. Hardware-Aware Token Pruning

Token pruning techniques are often designed without explicit consideration of hardware constraints [102]. However, real-world deployments require hardware-aware optimizations:

- **Parallelization-Friendly Pruning**: Some pruning methods introduce irregular sparsity patterns that are difficult to accelerate on GPUs and TPUs [103].
- **Energy Efficiency Considerations**: Future work should explore energy-aware pruning techniques that optimize power consumption on edge devices [104].
- **Integration with Specialized AI Accelerators**: Hardware accelerators, such as FPGAs and custom AI chips, may benefit from token pruning strategies tailored to their architectures.

Developing hardware-aware pruning techniques that align with modern AI accelerator architectures remains an important direction for future research [105].

6.5. Automated and Self-Supervised Pruning Methods

Current token pruning approaches often require task-specific tuning. Future research could explore more automated and self-supervised pruning techniques:

- **Meta-Learning for Pruning**: Can meta-learning be used to learn optimal pruning policies across multiple tasks [106]?
- **Self-Supervised Pruning**: Future models could learn pruning strategies in a self-supervised manner, reducing the need for labeled data [107].
- **Neural Architecture Search (NAS) for Pruning**: NAS techniques could be applied to discover optimal token pruning configurations dynamically.

These approaches could enable more flexible and generalizable pruning strategies that do not require extensive manual tuning [108].

6.6. Ethical Considerations in Token Pruning

As token pruning is integrated into real-world AI systems, ethical considerations must also be addressed:

- **Bias in Pruning Decisions**: If pruning disproportionately removes tokens related to certain demographics, it may introduce or amplify biases in NLP applications [109].
- **Explainability and Transparency**: Users of AI systems should be able to understand how pruning affects model decisions [110].
- **Fairness Across Languages and Dialects**: Many pruning methods are developed using English datasets, raising concerns about fairness in multilingual settings.

Future research should explore fairness-aware token pruning strategies that mitigate bias and improve model transparency [111].

 doi:10.20944/preprints202503.1577.v1

6.7. Summary and Future Outlook

Token pruning has emerged as a powerful tool for improving the efficiency of Transformer-based models [112]. However, several open challenges remain, including balancing accuracy and efficiency, extending pruning to multimodal models, optimizing for hardware constraints, and addressing ethical considerations. Future research will likely focus on developing more adaptive, automated, and hardware-aware pruning techniques to further enhance the scalability of large-scale AI systems. The next section concludes our survey, summarizing key insights and highlighting promising research directions in the field of token pruning [113].

7. Conclusion

Token pruning has emerged as a powerful and efficient technique for reducing the computational overhead of Transformer-based models while maintaining high performance across various NLP tasks. As deep learning models continue to scale in size and complexity, optimizing inference efficiency becomes a critical research challenge, especially for real-world applications requiring low latency, reduced memory consumption, and energy efficiency.

In this survey, we provided a comprehensive overview of token pruning, including its fundamental principles, various pruning methodologies, empirical evaluations, real-world applications, and open research challenges. Our key insights are summarized as follows:

- **Token Pruning Strategies**: Token pruning methods can be broadly categorized into static, dynamic, and hybrid approaches, each offering unique trade-offs in terms of computational savings and model accuracy retention.
- **Empirical Findings**: Empirical evaluations have demonstrated that well-designed pruning techniques can significantly reduce FLOPs and inference latency while maintaining high accuracy, particularly when dynamic and adaptive strategies are employed.
- **Real-World Applications**: Token pruning has been successfully integrated into various domains, including mobile NLP, real-time streaming applications, healthcare, and large-scale language model deployments, highlighting its practical relevance.
- **Open Challenges**: Despite its advantages, token pruning presents several research challenges, including preserving model generalization, optimizing pruning for multimodal architectures, ensuring hardware compatibility, and addressing ethical concerns such as fairness and bias.

Looking ahead, future research directions should focus on developing more adaptive and automated token pruning methods that dynamically adjust based on input characteristics and task-specific requirements. Additionally, integrating token pruning with other efficiency techniques, such as quantization, distillation, and neural architecture search (NAS), could further enhance the trade-offs between accuracy and computational cost.

Furthermore, as multimodal and generative AI models continue to gain traction, extending token pruning to these architectures will be an important area of exploration. Finally, ethical considerations, including bias mitigation and explainability in pruning decisions, should be given increased attention to ensure fair and transparent AI systems.

Token pruning represents a promising path toward making large-scale deep learning models more efficient and deployable in real-world settings. With ongoing advancements in AI optimization techniques and hardware-aware acceleration, token pruning is poised to play a vital role in shaping the future of efficient NLP and beyond.

References

1. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.
2. Gabriel Synnaeve Nicolas Carion, Francisco Massa. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2023.

 doi:10.20944/preprints202503.1577.v1

3. Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. Improving efficient neural ranking models with cross-architecture knowledge distillation, 2020.
4. Zhuyun Dai and Jamie Callan. Deeper text understanding for IR with contextual neural language modeling. *CoRR*, abs/1905.09217, 2019. URL http://arxiv.org/abs/1905.09217.
5. David Carmel, Doron Cohen, Ronald Fagin, Eitan Farchi, Michael Herscovici, Yoelle S Maarek, and Aya Soffer. Static index pruning for information retrieval systems. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50, 2001.
6. Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*, pages 620–640. Springer, 2022.
7. Huanrui Yang, Hongxu Yin, Maying Shen, Pavlo Molchanov, Hai Li, and Jan Kautz. Global vision transformer pruning with hessian-aware saliency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18547–18557, 2023.
8. Yuxin Fang, Bencheng Liao, Xinggang Wang, and Fang. You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems*, 34: 26183–26197, 2021.
9. S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 232–241, New York, NY, USA, 1994. Springer-Verlag New York, Inc. ISBN 0-387-19889-X. URL http://dl.acm.org/citation.cfm?id=188490.188561.
10. Emine Yilmaz, Javed A. Aslam, and Stephen Robertson. A new rank correlation coefficient for information retrieval. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, page 587–594, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581644. doi: 10.1145/1390334.1390435. URL https://doi.org/10.1145/1390334.1390435.
11. Massih-Reza Amini and Gaussier Eric. *Recherche d'Information - applications, modèles et algorithmes*. Algorithmes. Eyrolles, April 2013. URL https://hal.archives-ouvertes.fr/hal-00881257. I-XIX, 1-233.
12. Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10809–10818, 2022.
13. Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 113–122, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380379. doi: 10.1145/3404835.3462891. URL https://doi.org/10.1145/3404835.3462891.
14. Yongming Rao, Zuyan Liu, Wenliang Zhao, Jie Zhou, and Jiwen Lu. Dynamic spatial sparsification for efficient vision transformers and convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
15. Liu Yang Yi Tay, Dara Bahri. Sparse sinkhorn attention. *arXiv preprint arXiv:2002.11296*, 2020.
16. Haoran You, Yunyang Xiong, Xiaoliang Dai, Bichen Wu, Peizhao Zhang, Haoqi Fan, Peter Vajda, and Yingyan Celine Lin. Castling-vit: Compressing self-attention via switching towards linear-angular attention at vision transformer inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14431–14442, 2023.
17. Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.
18. Gary R Waissi. Network flows: Theory, algorithms, and applications, 1994.
19. Yukun Zheng, Zhen Fan, Yiqun Liu, Cheng Luo, Min Zhang, and Shaoping Ma. Sogou-qcl: A new dataset with click relevance label. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pages 1117–1120, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5657-2. doi: 10.1145/3209978.3210092. URL http://doi.acm.org/10.1145/3209978.3210092.
20. Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression, 2019. URL https://arxiv.org/abs/1902.09630.

 doi:10.20944/preprints202503.1577.v1

21. Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. Multi-stage document ranking with bert, 2019.
22. Joel Mackenzie, Zhuyun Dai, Luke Gallagher, and Jamie Callan. Efficiency implications of term weighting for passage retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1821–1824, 2020.
23. Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.
24. Dr. Hiteshwar Kumar Azad and A. Deepak. Query expansion techniques for information retrieval: a survey. *Inf. Process. Manag.*, 56:1698–1735, 2019.
25. Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34: 13937–13949, 2021.
26. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
27. Yassine Zniyed, Thanh Phuong Nguyen, et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
28. Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. Deeprank: A new deep architecture for relevance ranking in information retrieval. *CoRR*, abs/1710.05649, 2017. URL http://arxiv.org/abs/1710.05649.
29. Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pages 668–685. Springer, 2022.
30. Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, and Judy Hoffman. Hydra attention: Efficient attention with many heads. In Leonid Karlinsky, Tomer Michaeli, and Ko Nishino, editors, *Computer Vision – ECCV 2022 Workshops*, pages 35–49, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-25082-8.
31. Adam Berger and John Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, page 222–229, New York, NY, USA, 1999. Association for Computing Machinery. ISBN 1581130961. doi: 10.1145/312624.312681. URL https://doi.org/10.1145/312624.312681.
32. Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018.
33. Yixing Fan, Jiafeng Guo, Yanyan Lan, Jun Xu, Chengxiang Zhai, and Xueqi Cheng. Modeling diverse relevance patterns in ad-hoc retrieval. *CoRR*, abs/1805.05737, 2018. URL http://arxiv.org/abs/1805.05737.
34. Xiao Wang, Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. Pseudo-relevance feedback for multiple representation dense retrieval. In Faegheh Hasibi, Yi Fang, and Akiko Aizawa, editors, *ICTIR '21*, pages 297–306. ACM, 2021. doi: 10.1145/3471158.3472250. URL https://doi.org/10.1145/3471158.3472250.
35. Yingqi Qu Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering, 2020.
36. Mingjian Zhu, Yehui Tang, and Kai Han. Vision transformer pruning. *arXiv preprint arXiv:2104.08500*, 2021.
37. Kenji Doya Stefan Elfwing, Eiji Uchibe. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *arXiv preprint arXiv:1702.03118*, 2017.
38. Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. URL http://arxiv.org/abs/1405.0312. cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list.
39. Stéphane Aroca-Ouellette and Frank Rudzicz. On Losses for Modern Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4970–4981, Online, November 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-main.403.
40. Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. PoWER-BERT: Accelerating BERT inference via progressive word-vector elimination. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3690–3699. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/goyal20a.html.

 doi:10.20944/preprints202503.1577.v1

41. Soroush Abbasi Koohpayegani and Hamed Pirsiavash. Sima: Simple softmax-free attention for vision transformers. *arXiv preprint arXiv:2206.08898*, 2022.
42. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.
43. FirstName Alpher and FirstName Gamow. Can a computer frobnicate? In *CVPR*, pages 234–778, 2005.
44. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.
45. Ellen M. Voorhees and Donna K. Harman. *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press, 2005. ISBN 0262220733.
46. Zhuyun Dai and Jamie Callan. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687*, 2019.
47. Deepthi Karkada Aishwarya Bhandare, Vamsi Sripathi. Efficient 8-bit quantization of transformer neu- ral machine language translation model. *arXiv preprint arXiv:1906.00532*, 2019.
48. Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Jakob Verbeek, and Hervé Jégou. Three things everyone should know about vision transformers. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 497–515, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20053-3.
49. Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert, 2019.
50. Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. On the robustness of vision transformers to adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7838–7847, 2021.
51. Jia Deng, R. Socher, Li Fei-Fei, Wei Dong, Kai Li, and Li-Jia Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, volume 00, pages 248–255, 06 2009. doi: 10.1109/CVPR.2009.5206848. URL https://ieeexplore.ieee.org/abstract/document/5206848/.
52. Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang. Soft: Softmax-free transformer with linear complexity. *Advances in Neural Information Processing Systems*, 34:21297–21309, 2021.
53. Tiancheng Zhao, Xiaopeng Lu, and Kyusong Lee. Sparta: Efficient open-domain question answering via sparse transformer matching retrieval, 2020.
54. Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19175–19186, 2023.
55. Yue Cao Ze Liu, Yutong Lin. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.
56. Bohan Zhuang, Jing Liu, Zizheng Pan, Haoyu He, Yuetian Weng, and Chunhua Shen. A survey on efficient training of transformers. pages 6823–6831, 08 2023. doi: 10.24963/ijcai.2023/764.
57. Xianing Chen, Qiong Cao, Yujie Zhong, Jing Zhang, Shenghua Gao, and Dacheng Tao. Dearkd: data-efficient early knowledge distillation for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12052–12062, 2022.
58. Wenyu Lv, Shangliang Xu, Yian Zhao, Guanzhong Wang, Jinman Wei, Cheng Cui, Yuning Du, Qingqing Dang, and Yi Liu. Detrs beat yolos on real-time object detection. *arXiv preprint arXiv:2304.08069*, 2023.
59. Ziheng Wang, Jeremy Wohlwend, and Tao Lei. Structured pruning of large language models. *arXiv preprint arXiv:1910.04732*, 2019.
60. Tharun Medini, Beidi Chen, and Anshumali Shrivastava. {SOLAR}: Sparse orthogonal learned and random embeddings. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=fw-BHZ1KjxJ.
61. FirstName Alpher and FirstName Fotheringham-Smythe. Frobnication revisited. *Journal of Foo*, 13(1): 234–778, 2003.

 doi:10.20944/preprints202503.1577.v1

62. Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwikj. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=zeFrfgyZln.
63. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. URL http://arxiv.org/abs/1301.3781.
64. Gyuwan Kim and Kyunghyun Cho. Length-adaptive transformer: Train once with length drop, use anytime with search. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6501–6511, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.508. URL https://aclanthology.org/2021.acl-long.508.
65. Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021.
66. Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. 88, 01 2000.
67. Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6399–6408, 2019.
68. A Kolesnikov A Dosovitskiy, L Beyer. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929v2*, 2021.
69. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.
70. Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020.
71. Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. *Click Models for Web Search*. Morgan & Claypool, 2015. ISBN 9781627056489. doi: 10.2200/S00654ED1V01Y201507ICR043.
72. Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications, 2023. URL https://arxiv.org/abs/2304.07288.
73. Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *arXiv preprint arXiv:2005.10242*, 2020.
74. Bhaskar Mitra and Nick Craswell. An introduction to neural information retrieval. *Foundations and Trends in Information Retrieval*, pages 1–117, April 2018. URL https://www.microsoft.com/en-us/research/publication/introduction-neural-information-retrieval/.
75. Fanfan Liu, Haoran Wei, and Zhao. Wb-detr: Transformer-based detector without backbone. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2959–2967, 2021. doi: 10.1109/ICCV48922.2021.00297.
76. Li Yuan Zihang Jiang, Qibin Hou. All tokens matter: Token labeling for training better vision transformers. *arXiv preprint arXiv:2104.10858*, 2021.
77. Yassine Zniyed, Thanh Phuong Nguyen, et al. Efficient tensor decomposition-based filter pruning. *Neural Networks*, 178:106393, 2024.
78. Jinnian Zhang, Houwen Peng, Kan Wu, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Minivit: Compressing vision transformers with weight multiplexing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12145–12154, 2022.
79. Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.
80. Kenton Lee J Devlin, M Chang. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
81. Paul Pu Liang, Manzil Zaheer, Yuan Wang, and Amr Ahmed. Anchor & transform: Learning sparse embeddings for large vocabularies, 2020.
82. Max Raphael Sobroza, Tales Marra, Deok-Hee Kim-Dufor, and Claude Berrou. Sparse associative memory based on contextual code learning for disambiguating word senses. *arXiv preprint arXiv:1911.06415*, 2019.
83. Antonio Mallia, Michal Siedlaczek, Joel Mackenzie, and Torsten Suel. Pisa: performant indexes and search for academia. *Proceedings of the Open-Source IR Replicability Challenge*, 2019.
84. Kavindu Chamith Hans Thisanke, Chamli Deshan. Semantic segmentation using vision transformers: A survey. *arXiv preprint arXiv:2305.03273*, 2023.

 doi:10.20944/preprints202503.1577.v1

85. Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.
86. Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.
87. Jing Liu, Zizheng Pan, Haoyu He, Jianfei Cai, and Bohan Zhuang. Ecoformer: Energy-saving attention with linear complexity, 2023.
88. Raphael Shu and Hideki Nakayama. Compressing word embeddings via deep compositional code learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJRZzFlRb.
89. Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJlnC1rKPB.
90. Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.
91. Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021.
92. Ye Zhang, Md. Mustafizur Rahman, Alex Braylan, Brandon Dang, Heng-Lu Chang, Henna Kim, Quinten McNamara, Aaron Angert, Edward Banner, Vivek Khetan, Tyler McDonnell, An Thanh Nguyen, Dan Xu, Byron C. Wallace, and Matthew Lease. Neural information retrieval: A literature review. *CoRR*, abs/1611.06792, 2016. URL http://arxiv.org/abs/1611.06792.
93. Ross Girshick Yanghao Li, Hanzi Mao. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022.
94. Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022.
95. Ehud D Karnin. A simple procedure for pruning back-propagation trained neural networks. *IEEE transactions on neural networks*, 1(2):239–242, 1990.
96. Marcos V Conde, Ui-Jin Choi, Maxime Burchi, and Radu Timofte. Swin2SR: Swinv2 transformer for compressed image super-resolution and restoration. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2022.
97. David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3: 993–1022, March 2003. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=944919.944937.
98. Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. Repbert: Contextualized text embeddings for first-stage retrieval, 2020.
99. Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1253–1256, 2017.
100. Yan Xiao, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng. Beyond precision: A study on recall of initial retrieval with neural representations. *CoRR*, abs/1806.10869, 2018. URL http://arxiv.org/abs/1806.10869.
101. Xuran Pan, Tianzhu Ye, Zhuofan Xia, Shiji Song, and Gao Huang. Slide-transformer: Hierarchical vision transformer with local self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2082–2091, 2023.
102. Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, 2016.
103. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.
104. Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.

 doi:10.20944/preprints202503.1577.v1

105. Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.
106. Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 36, pages 2071–2081, 2022.
107. Ziyao Guo, Haonan Yan, Hui Li, and Xiaodong Lin. Class attention transfer based knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11868–11877, 2023.
108. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
109. Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.
110. Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers for image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 12–21, 2023.
111. Charles R. Harris, K. Jarrod Millman, St'efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern'andez del R'10, Mark Wiebe, Pearu Peterson, Pierre G'erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.
112. Wei Yang, Kuang Lu, Peilin Yang, and Jimmy Lin. Critically examining the "neural hype". *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul 2019. doi: 10.1145/3331184.3331340. URL http://dx.doi.org/10.1145/3331184.3331340.
113. Sucheng Ren, Zhengqi Gao, Tianyu Hua, Zihui Xue, Yonglong Tian, Shengfeng He, and Hang Zhao. Co-advise: Cross inductive bias distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 16773–16782, 2022.